CLAIMS

We claim:

1. A method for performing a table look-up operation on a first table having N entries, said first table having a first data field of table index values and a second data field of computed table values, said table index values of said first data field being separated by one or more intervals, said method comprising:

generating a second table having $k$N entries based on said first table, where $k$ is an integer and $k$N has a value being the power of 2, comprising:

generating a first data field including a plurality of table index values being selected from a predefined range, said plurality of table index values having a second interval derived from a first interval selected from said one or more intervals of said first table, said second interval being a value represented by an $n$-bit binary number;

generating a second data field including a plurality of computed table values derived from said computed table values of said first table;

computing an index value z;

extracting address bits from said index value z, said address bits being data bits more significant than the ($n$-1)th bit of said index value z; and

addressing said second table using said address bits.

2. The method of claim 1, wherein said second interval is given by $2^{\lfloor \log_2(z_I) \rfloor}$, where $z_I$ denotes said first interval and the notation $\lfloor x \rfloor$ represents the largest integer value not greater than $x$; and said second interval is represented by said $n$-bit binary number where $n = \lfloor \log_2(z_I) \rfloor$.

3. The method of claim 1, wherein said extracting address bits from said index value z comprises:

extracting m bits of address bits, where m is the number of bits required to represent the value *k*N in binary number.

4. The method of claim 1, wherein said second table has 2N entries.

5. The method of claim 4, wherein said extracting address bits from said index value z comprises:

extracting m bits of address bits, where m is the number of bits required to represent the value 2N in binary number.

6. The method of claim 1, wherein said generating a second data field including a plurality of computed table values derived from said computed table values of said first table comprises:

generating said computed table values of said second table by linear interpolation of said computed table values of said first table.

7. The method of claim 1, wherein said generating a second data field including a plurality of computed table values derived from said computed table values of said first table comprises:

computing said computed table values of said second table using a function used for computing said computed table values of said first table.

8. The method of claim 1, further comprising:

obtaining a first computed table value from said plurality of computed table values in said second data field of said second table using said address bits.

9. A circuit for decoding input data, comprising:

a decoder implementing the maximum a posteriori probability decoding algorithm, said decoder using a first table for computing the function $\log(e^{x_1}+e^{x_2})$ or $\ln(e^{x_1}+e^{x_2})$ where $x_1$ and $x_2$ are first and second argument values, each derived from said input data, said first table having N entries and storing a first data field including a plurality of table index values and a second data field including a plurality of computed table

values corresponding to said plurality of table index values, said plurality of table index values are selected from a predefined range of $|x_1 - x_2|$ argument values, said table index values of said first data field are separated by one or more intervals, and said plurality of computed table values are computed based on the equation $\log(1+e^{-|x_1-x_2|})$ or $\ln(1+e^{-|x_1-x_2|})$ for each of said $|x_1 - x_2|$ argument values selected for said table index values; and

said decoder using a second table having *k*N entries and storing a first data field including a plurality of table index values and a second data field including a plurality of computed table values corresponding to said plurality of table index values;

wherein said plurality of table index values of said second table are selected from said predefined range of $|x_1 - x_2|$ argument values and have a second interval derived from a first interval selected from said one or more intervals of said first table, said second interval being a value represented by an *n*-bit binary number; and said plurality of computed table values of said second table are derived from said computed table values of said first table; and

wherein said second table is addressed by using address bits in an index value z and said address bits are data bits more significant than the (*n*-1)th bit of said index value z.

10. The circuit of claim 9, wherein said plurality of table index values of said first table and said plurality of said computed table values of said first table are scaled by a first scaling factor.

11. The circuit of claim 9, wherein said address bits have *m* bits, where *m* is the number of bits required to represent the value *k*N in binary number.

12. The circuit of claim 9, wherein said second table has 2N entries.

13. The circuit of claim 12, wherein said address bits have *m* bits, where *m* is the number of bits required to represent the value 2N in binary number.

14. The circuit of claim 9, wherein said decoder computes the function $\log(e^{x_1} + e^{x_2})$ or $\ln(e^{x_1} + e^{x_2})$ for said index value z by obtaining a first computed table value from said plurality of computed table values in said second table using said address bits.

15. The circuit of claim 14, wherein said first computed table value is added to the greater of said first argument value $x_1$ and said second argument value $x_2$.

16. The circuit of claim 9, wherein said computed table values of said second table are computed by linear interpolation of said computed table values of said first table.

17. The circuit of claim 9, wherein said computed table values of said second table are computed using a function used for computing said computed table values of said first table.

18. A method in a decoder applying the maximum a-posteriori probability algorithm for computing the function $\log(e^{x_1} + e^{x_2})$ or $\ln(e^{x_1} + e^{x_2})$ for a first argument value $x_1$ and a second argument value $x_2$, comprising:

generating a first table having N entries, comprising:

generating a first data field including a plurality of table index values being selected from a predefined range of $|x_1-x_2|$ argument values, said plurality of table index values of said first data field being separated by one or more intervals;

generating a second data field including a plurality of computed table values computed based on the equation $\log(1+e^{-|x_1-x_2|})$ or $\ln(1+e^{-|x_1-x_2|})$ for each of said $|x_1-x_2|$ argument values selected for said table index values;

generating a second table having *k*N entries, comprising:

generating a first data field including a plurality of table index values being selected from said predefined range of $|x_1-x_2|$ argument values, said plurality of table index values having a second interval derived from a first interval selected from said one or more intervals of said first table, said second interval being a value represented by an $n$-bit binary number;

generating a second data field including a plurality of computed table values derived from said computed table values of said first table;

computing an index value z;

extracting address bits from said index value z, said address bits being data bits more significant than the ($n$-1)th bit of said index value z; and

addressing said second table using said address bits.

19. The method of claim 18, wherein said generating a first table further comprises:

scaling said table index values of said first table by a first scaling factor; and

scaling said computed table values of said first table by said first scaling factor.

20. The method of claim 18, wherein said extracting address bits from said index value z comprises:

extracting m bits of address bits, where m is the number of bits required to represent the value $k$N in binary number.

21. The method of claim 18, wherein said second table has 2N entries.

22. The method of claim 21, wherein said extracting address bits from said index value z comprises:

extracting m bits of address bits, where m is the number of bits required to represent the value 2N in binary number.

23. The method of claim 18, further comprising:

obtaining a first computed table value from said plurality of computed table values in said second table using said address bits.

24. The method of claim 23, further comprising:

determining a greater of said first argument value $x_1$ and said second argument value $x_2$; and

adding said first computed table value to said greater of said first argument value $x_1$ and said second argument value $x_2$.

25. The method of claim 18, wherein said generating a second data field including a plurality of computed table values derived from said computed table values of said first table comprises:

generating said computed table values of said second table by linear interpolation of said computed table values of said first table.

26. The method of claim 18, wherein said generating a second data field including a plurality of computed table values derived from said computed table values of said first table comprises:

computing said computed table values of said second table using a function used for computing said computed table values of said first table.